

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-358102

(43)Date of publication of application : 13.12.2002

(51)Int.Cl.

G05B 19/05

(21)Application number : 2001-164604

(71)Applicant : OKUMA CORP

(22)Date of filing : 31.05.2001

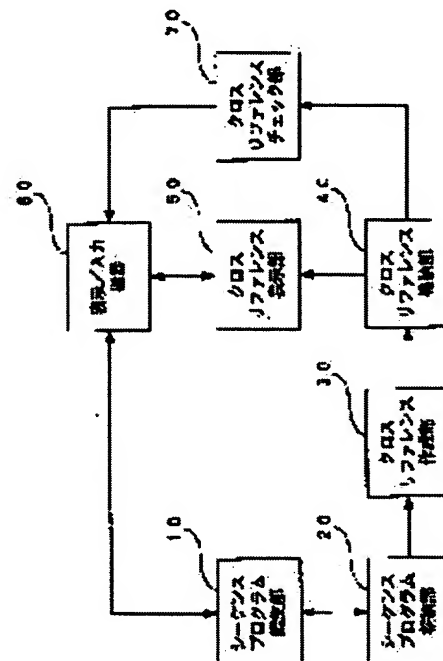
(72)Inventor : NAKANO EIZO  
TAUCHI HIDEKAZU

## (54) PROGRAMMING DEVICE FOR PROGRAMMABLE LOGIC CONTROLLER

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a programming device of a programmable logic controller for executing a sequence program by allocating it to a plurality of tasks whose start cycles are different, which has a function of detecting that the writing of bit data is generated from those tasks whose cycles are different to the same address, and informing a program preparer of the result.

**SOLUTION:** In this programming device of a programmable logic controller for executing a sequence program by allocating it to a plurality of tasks whose start cycles are different, it is detected that the writing of bit data in the same address is present so as to be overlapped in the tasks whose cycles are different in the sequence program, and the detected result is displayed together with the task number, program unit number, and the address and bit position of the bit data to be written.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2002-358102

(P2002-358102A)

(43)公開日 平成14年12月13日(2002.12.13)

(51)Int.Cl.

識別記号

F I

テーマコード(参考)

G 0 5 B 19/05

G 0 5 B 19/05

A 5 H 2 2 0

審査請求 未請求 請求項の数3 O L (全 6 頁)

(21)出願番号 特願2001-164604(P2001-164604)

(22)出願日 平成13年5月31日(2001.5.31)

(71)出願人 000149066

オークマ株式会社

愛知県丹羽郡大口町下小口五丁目25番地の  
1

(72)発明者 中野 栄造

愛知県丹羽郡大口町下小口五丁目25番地の  
1 オークマ株式会社内

(72)発明者 田内 秀和

愛知県丹羽郡大口町下小口五丁目25番地の  
1 オークマ株式会社内

(74)代理人 100078776

弁理士 安形 雄三 (外2名)

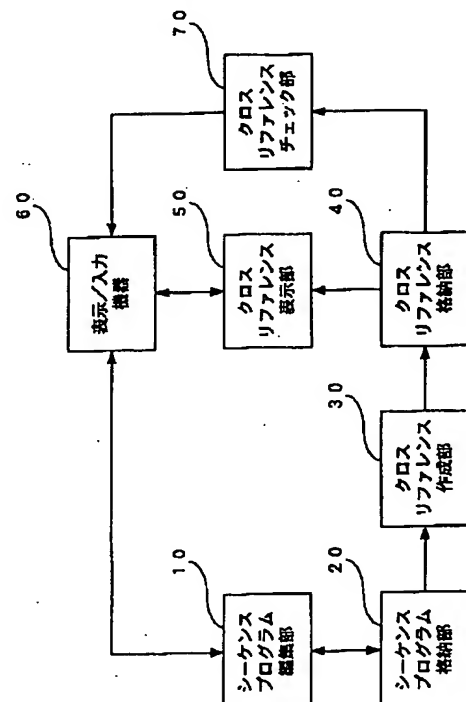
最終頁に続く

(54)【発明の名称】 プログラマブル・ロジック・コントローラのプログラミング装置

(57)【要約】

【課題】シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラにおいて、周期の異なるタスクから同一のアドレスに対して、ビットデータの書き込みが発生することを検出し、プログラム作成者に通知する機能を有するプログラマブル・ロジック・コントローラのプログラミング装置を提供する。

【解決手段】シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラのプログラミング装置において、前記シーケンスプログラムの中に同一アドレスに対するビットデータ書き込みが複数周期のタスクに重複して存在する場合を検出し、前記検出結果を、タスク番号、プログラム単位番号、書き込むビットデータのアドレスとビット位置を併せて表示する。



1

**【特許請求の範囲】**

【請求項1】シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラのプログラミング装置において、前記シーケンスプログラムの中に同一アドレスに対するビットデータ書き込みが複数周期のタスクに重複して存在する場合を検出し、前記検出結果を、タスク番号、プログラム単位番号、書き込むビットデータのアドレス及びビット位置を併せて表示することを特徴とするプログラマブル・ロジック・コントローラのプログラミング装置。

【請求項2】更にタスク名称が表示され、前記プログラム単位がワークシート番号及びワークシート名称である請求項1に記載のプログラマブル・ロジック・コントローラのプログラミング装置。

【請求項3】更に読み出し／書き込みの区別が表示される請求項2に記載のプログラマブル・ロジック・コントローラのプログラミング装置。

**【発明の詳細な説明】****【0001】**

20

【発明の属する技術分野】本発明は、シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラ（以下、単に「PLC」と略す）のプログラミング装置に関する。

**【0002】**

【従来の技術】シーケンスプログラムの実行を汎用CPUで行う方式、いわゆるソフトウェアPLCが実用化され普及してきている。汎用CPUにおいては、ビットデータを書き込む際に、下記3ステップ（1）～（3）の30処理が必要となる。

（1）メモリから1バイト（CPUによっては2又は4バイト）データを読み出す。

（2）AND又はORのデータでマスクを取り、目的のビットをON/OFFする。

（3）メモリに1バイト（CPUによっては2又は4バイト）データを書き戻す。

【0003】ところが、起動周期の異なる複数のタスクでシーケンスプログラムを構成するマルチタスクシステムの場合、タスクAからビットデータを書き込むために40上記ステップ（2）を実行している間に割り込みが入り、タスクBに切り替わって、同一アドレスに対してビットデータを書き込みが行われる場合がある。タスクBが終了した直後に元のタスクAに戻って、上記ステップ（3）のデータ書き込みが発生すると、タスクBで書き込んだはずのビットデータが失われてしまうという問題があった。

【0004】上記従来のPLCのプログラミング装置では、このような複数周期のタスクでシーケンスプログラムを構成するマルチタスクシステムの場合、同一アドレ

2

スに対するビットデータの書き込みで干渉が発生してしまう。このため、最近では、データ不整合が起きることを防止する必要から、CPUが持つハードウェア割り込み禁止命令を使用することが提案されている。つまり、シーケンスプログラムをターゲットCPUで動作する機械語命令に変換する時、ビットデータの書き込み中にタスクが切り替わらないように割り込み禁止命令を追加するのである。ビットデータをメモリに書き戻した後で、割り込み許可命令を実行してタスクの切り替えを可能とすることで、ビットデータの書き込み干渉を防止できる。その結果、ビットデータの書き込みには、以下のような5ステップ（a）～（e）の処理が必要となる。

（a）割り込み禁止命令を実行する。

（b）メモリから1バイト（CPUによっては2又は4バイト）データを読み出す。

（c）AND又はORのデータでマスクを取り、目的のビットをON/OFFする。

（d）メモリに1バイト（CPUによっては2又は4バイト）データを書き戻す。

（e）割り込み許可命令を実行する。

【0005】また、割り込みレベルを複数持つことができるCPUでは、ビットデータを書き込む時に、割り込みレベルをタスク切り替えに使用する割り込み以上に上げておくことで、割り込み禁止命令を使用した場合と同等の効果が期待できる。割り込み禁止命令と、割り込みレベルのどちらを使用するかは、シーケンスプログラムを実行するターゲットCPU毎に判断し、適している方を採用すれば良い。

【0006】しかし、このような割り込み禁止命令を使用する方式では、プログラム実行の効率が悪くなるという問題がある。

【0007】また、シーケンスプログラムを一度C言語に代表される汎用プログラミング言語に変換し、ターゲットCPU用のCコンパイラで実行プログラム化する方式のプログラミング装置では、Cコンパイラ自体がビットデータを書き込む処理について、CPUの割り込み禁止命令を組み込む機能をサポートしている必要がある。

【0008】さらに、ターゲットCPUによっては、割り込み禁止命令がない場合や割り込みレベルを変更できない場合も存在する。

【0009】そこで別の対策として、ビットが干渉しないようにタスク毎にビットデータを書き込むアドレスを分けることが提案されている。即ち、ワークメモリ等のPLC内部で使用する変数については、アドレスを持たないラベルでプログラミングを行い、プログラミング装置がアドレスを自動で割り付けることが可能である。アドレスを自動割り付けする際に、タスク毎にビットデータを書き込むアドレスが重複しないようにすることで、ビットデータ書き込み干渉を防止できるようにしている。

3

【0010】しかし、I/O（入出力）などPLC外部とインタフェースをとる変数については、プログラミング装置はアドレスを自動的に割り付けることができない。そこで、変数のクロスリファレンスデータを基に、プログラム作成者がタスク間でビットデータの書き込みに干渉が発生しないようにチェックをする必要がある。

【0011】図4は、従来のプログラミング装置におけるクロスリファレンスデータ表示機能の一実施例を示すブロック構成図である。シーケンスプログラム編集部10は、シーケンスプログラム格納部20よりシーケンス10プログラムを読み出し、表示／入力機器60におけるCRT等の表示装置にラダー図、SFC（シーケンシャル・ファンクション・チャート）等の形式で表示する。プログラム作成者は、表示／入力機器60におけるキーボード、マウス等の入力装置を使用してシーケンスプログラムを編集する。編集されたシーケンスプログラムは、再びシーケンスプログラム格納部20に格納される。

【0012】クロスリファレンス作成部30は、シーケンスプログラム格納部20よりシーケンスプログラムを読み出し、各変数毎に使用されているタスク番号とプログラム単位番号、読み出し／書き込みの区別についてまとめたクロスリファレンスデータを作成する。作成されたクロスリファレンスデータは、クロスリファレンス格納部40に格納される。

【0013】クロスリファレンス表示部50は、クロスリファレンス格納部40よりクロスリファレンスデータを読み出し、表示／入力機器60におけるCRT等の表示装置に表示する。表示する順序は、変数のアドレス順又はラベル順で切り換え可能である。また、クロスリファレンスデータはデータサイズを圧縮するために、タスク番号とプログラム単位番号のように数値データとして持っているが、表示する際には、これらをタスク名称とプログラム単位名称に変換して表示することができる。タスク番号及び名称、プログラム単位番号及び名称のように両方を表示することも可能である。

【0014】図5は、クロスリファレンスデータを表示した画面例を示している。左欄から順に「アドレス」欄に変数のアドレス210を、「ビット」欄にビット位置220を、「ラベル」欄にラベル名称230を、「タスク」欄にタスク番号240及びタスク名称250を、40「ワークシート」欄にプログラムを作成する単位であるワークシート番号260及びワークシート名称270を、「R/W」欄に読み出し（R）／書き込み（W）の区別280をそれぞれ表示している。画面は、画面右側のスクロールバーによって上下にスクロール可能である。

【0015】プログラム作成者は、このようなクロスリファレンスデータを画面に表示（又は用紙に印刷）し、同一アドレスの各ビットについて書き込みを行なっているタスクを洗い出し、タスクの起動周期が異なっていないかをチェックする。

4

【0016】

【発明が解決しようとする課題】 上述した従来のプログラミング装置においては、I/Oなどアドレスを自動で振り当てるのが困難なデータについては、プログラム作成者が事前にタスクの起動周期に注意してアドレスを割り振るか、事後にビットデータの書き込みに関するクロスリファレンスデータを画面又は印刷結果を基にチェックし、同一アドレスのビットデータ書き込みについては、割り付けるタスクの周期が同一になるようにアドレスを振り当て直す必要がある。このことは、プログラム作成者に多大な労力と時間を強いるものである上に、チェック漏れが発生した場合には、ビットデータ書き込みの干渉によりシーケンスプログラムが予期しない動作をする可能性があった。

【0017】本発明は上述のような事情からなされたものであり、本発明の目的は、シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラにおいて、周期の異なるタスクから同一のアドレスに対して、ビットデータの書き込みが発生することを検出し、プログラム作成者に通知する機能を有するプログラマブル・ロジック・コントローラのプログラミング装置を提供することにある。

【0018】

【課題を解決するための手段】 本発明は、シーケンスプログラムを起動周期の異なる複数のタスクに割り付けて実行するプログラマブル・ロジック・コントローラのプログラミング装置に関するものであり、本発明の上記目的は、前記シーケンスプログラムの中に同一アドレスに対するビットデータ書き込みが複数周期のタスクに重複して存在する場合を検出し、前記検出結果を、タスク番号、プログラム単位番号、書き込むビットデータのアドレスとビット位置を併せて表示することにより達成される。

【0019】

【発明の実施の形態】 以下、本発明の実施例を、図面を参照して具体的に説明する。図1は、本発明のプログラミング装置におけるクロスリファレンスデータ表示機能の一実施例を、図4に対応させて示すブロック構成図であり、同一構成箇所には同一符号を付して説明を省略する。本発明では、クロスリファレンス格納部40及び表示／入力機器60に関連するクロスリファレンスチェック部70が追加されている。

【0020】追加されたクロスリファレンスチェック部70の動作を、図2に示すフローチャートで説明する。クロスリファレンスチェック部70は、先ずクロスリファレンス格納部40から同一アドレスのクロスリファレンスデータを全て読み出す（ステップS100）。ここで言う同一アドレスとは、シーケンスプログラムから見た論理的なアドレスではなく、ターゲットCPUがメモ

5

りにアクセスする際のアドレスのことである。例えばシーケンスプログラムでは、ビット0～ビット15までの2バイトを1まとめにして同一のアドレスとして扱っている場合があるが、メモリに1バイト単位でアクセスするターゲットCPUにおいては、ビット0～ビット7と、ビット8～ビット15は異なったアドレスを使用する。このため、これらを別々のアドレスとして分類する必要がある。

【0021】次に、クロスリファレンスチェック部70は、読み出したクロスリファレンスデータの中でビット10データ書き込みに関するものだけを抽出し、割り付けタスクの周期をチェックする(ステップS110)。タスクの周期はクロスリファレンスデータのタスク番号を基に、図示しない別のテーブルを参照することで決まるものとする。

【0022】このようにして求めた割り付けタスクの周期が同一であるかどうかを判定し(ステップS120)、同一でない場合は表示/入力機器60のCRTなどの表示装置の画面に警告表示を行なう(ステップS130)。

【0023】その後、若しくは上記ステップS120においてタスク周期が同一と判定された場合には、ステップS100からステップS130までの処理を、クロスリファレンス格納部40に格納されているクロスリファレンスデータの全アドレスについて、終了するまで繰り返す(ステップS140)。

【0024】図3は、クロスリファレンスデータを基にビット書き込みの干渉を警告表示(クロスリファレンス干渉チェック表示)した画面の一例である。左欄から順に「アドレス」欄に変数のアドレス210を、「ビット」欄にビット位置220を、「ラベル」欄にラベル名称を、「タスク」欄にタスク番号240及びタスク名称270を、「ワークシート」欄にプログラムを作成する単位であるワークシート番号260及びワークシート名称270を、「R/W」欄に読み出し(R)/書き込み(W)の区別280をそれぞれ表示している。図5の場合と異なり、同一アドレスに書き込みを行なう全ビットについて一覧で表示している。また、タスク番号と名称、プログラムを作成する単位であるワークシート番号と名称が同一画面に表示されている。よって、図5のクロスリファレンスデータ表示画面と同等の情報を見落と

6

すことなく、より整理されたかたちで見ることが出来る。画面はスクロールバーにて、上下にスクロール可能である。

#### 【0025】

【発明の効果】以上のように本発明によれば、PLCのプログラミング装置において、周期の異なる複数のタスクから同一アドレスに対してビットデータの書き込みが発生することを判定し、クロスリファレンスデータから必要な情報を抽出して整理した状態で警告表示を行なう。このため、プログラム作成者がビットデータの書き込みに関するクロスリファレンスデータを、画面又は印刷結果を基に直接チェックする必要がなくなる。このことは、プログラム作成者の負担を軽減する上で効果があるばかりでなく、プログラム作成者のチェック漏れにより、同一アドレスに対するビットデータの書き込みで干渉が発生して、データ不整合が起こることを未然に防ぐことが可能になる。つまり、シーケンスプログラムの品質向上にも効果が大である。

#### 【図面の簡単な説明】

【図1】本発明のプログラミング装置におけるクロスリファレンスデータ表示機能の一実施例を示すブロック構成図である。

【図2】本発明のプログラミング装置におけるクロスリファレンスチェック部の動作を説明するフローチャートである。

【図3】本発明のプログラミング装置におけるクロスリファレンスデータによるビットデータ書き込み干渉チェック画面の一例を示す図である。

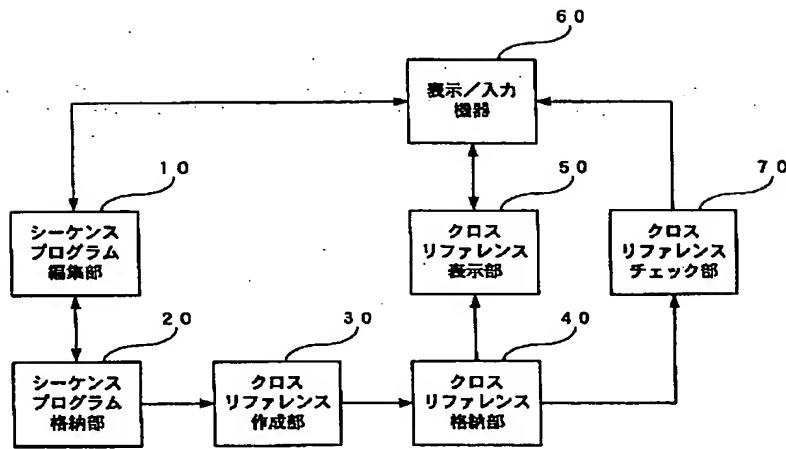
【図4】従来のプログラミング装置におけるクロスリファレンスデータ表示機能の一実施例を示すブロック構成図である。

【図5】従来のプログラミング装置におけるクロスリファレンスデータ表示画面の一例を示す図である。

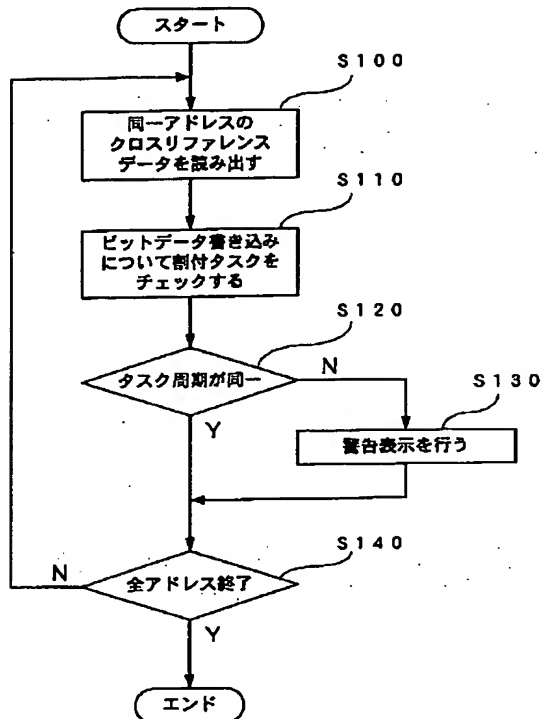
#### 【符号の説明】

- |    |                |
|----|----------------|
| 10 | シーケンスプログラム編集部  |
| 20 | シーケンスプログラム格納部  |
| 30 | クロスリファレンス作成部   |
| 40 | クロスリファレンス格納部   |
| 50 | クロスリファレンス表示部   |
| 60 | 表示/入力機器        |
| 70 | クロスリファレンスチェック部 |

【図 1】



【図 2】

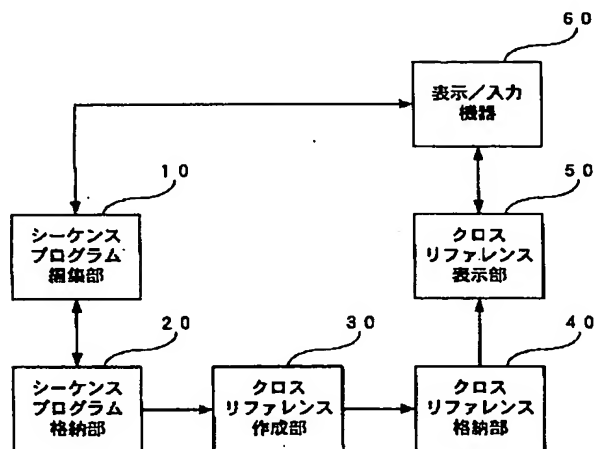


【図 3】

| アドレス  | ビット | ラベル    | タスク     | ワークシート         | R/W |
|-------|-----|--------|---------|----------------|-----|
| Q.021 | 0   | qSOL-A | 001 PLH | 0021 ATC_CNTRL | W   |
|       | 3   | qSOL-B | 001 PLH | 0021 ATC_CNTRL | W   |
|       | 4   | qCLMPS | 010 PLM | 0105 CLMP_CHK  | W   |
|       | 5   | qPSPD  | 020 PLT | 0232 PANEL_OUT | W   |
|       | 6   | qALMO  | 020 PLT | 0232 PANEL_OUT | W   |
|       | 7   | qRUNG  | 020 PLT | 0232 PANEL_OUT | W   |
|       |     |        |         |                |     |
| Q.024 | 0   | qPWOFF | 010 PLW | 0106 PWR_CNTRL | W   |
|       | 1   | qPWON  | 010 PLW | 0106 PWR_CNTRL | W   |
|       | 2   | qOPON  | 010 PLW | 0106 PWR_CNTRL | W   |
|       | 3   | qBKRLS | 010 PLW | 0107 BRK_CNTRL | W   |
|       | 7   | qBLOFF | 020 PLT | 0232 PANEL_OUT | W   |
|       |     |        |         |                |     |
|       |     |        |         |                |     |

Address range: 210 to 280. Bit range: 0 to 7. Task range: 001 PLH to 020 PLT. Worksheet range: 0021 ATC\_CNTRL to 0232 PANEL\_OUT.

【図 4】



【図5】

| クロスリファレンス表示 |     |        |         |                |     |
|-------------|-----|--------|---------|----------------|-----|
| アドレス        | ビット | ラベル    | タスク     | ワークシート         | R/W |
| Q.021       | 0   | qSOL-A | 001 PLH | 0021 ATC_CNTRL | W   |
|             |     |        | 001 PLH | 0031 TIMR_CHK  | R   |
|             |     |        | 010 PLM | 0101 ATC_CHK   | R   |
|             |     |        | 010 PLM | 0111 SPINDLE01 | R   |
|             |     |        | 010 PLM | 0112 SPINDLE02 | R   |
|             |     |        | 020 PLT | 0231 PANEL_CHK | R   |
| Q.021       | 3   | qSOL-B | 001 PLH | 0021 ATC_CNTRL | W   |
|             |     |        | 001 PLH | 0031 TIMR_CHK  | R   |
|             |     |        | 020 PLT | 0231 PANEL_CHK | R   |
| Q.021       | 4   | qCLMPS | 010 PLM | 0105 CLMP_CHK  | W   |

210 220 230 240 250 260 270 280

フロントページの続き

Fターム(参考) 5H220 BB12 CC05 EE08 HH01 JJ12  
JJ28 JJ53 KK08 MM03